IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of GORTHY et al.		Confirmation No.:	9237
Serial No.:	09/942,834	Examiner:	H. PHILLIPS
Filed:	08/29/2001	Art Unit:	2151
FOR:	SYSTEM AND METHOD FOR GENERATING A CONFIGURATION SCHEMA		

Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

37 C.F.R. §41.37(c) SUBSTITUTE APPEAL BRIEF

Sir:

In response to the Notice of Non-Compliant Appeal Brief dated September 19, 2007 Applicant hereby provides a Substitute Appeal Brief to replace the Appeal Brief filed January 10, 2007 and Substitute Appeal Brief filed June 12, 2007. Applicants hereby appeal from the Final Action of January 3, 2006 and the Advisory Action of May 22, 2006. The Notice of Appeal and a Pre-Appeal Brief Request for Review was filed on June 5, 2006.

REAL PARTY IN INTEREST

The real party in interest in this appeal is Intelliden Inc., as the assignee.

RELATED APPEALS AND INTERFERENCES

U.S. Application No. 09/942,833 entitled SYSTEM AND METHOD FOR MODELING A NETWORK DEVICE'S CONFIGURATION is also assigned to Intelliden Inc. and is also

currently under appeal.

U.S. Application No. 09/730,682 entitled NETWORK OPERATING SYSTEM DIRECTORY is also assigned to Intelliden Inc. and is also currently under appeal.

STATUS OF CLAIMS

Claims 6-12, 16, 24-25, and 27 are pending, stand as rejected and are being appealed. Claims 6 and 24 are independent. Claims 1-5, 13-15, 17-23, 26 and 28-29 are cancelled. The appendix includes a true copy of all pending claims. No claims have been allowed.

STATUS OF AMENDMENTS

The Advisory Action indicates that amendments to claims 6, 11 and 24 that were made after final rejection have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 6

Although several embodiments of the present invention are disclosed in the specification, Figures 3 and 4 and the supporting text provides a good summary of embodiments which are exemplary of the subject matter defined by independent claim 6 and at least a portion of the dependent claims. The main text describing Figures 3 and 4 is located at paragraphs [0021]-[0024] (Page 9, line 3 to Page 12, line 2) of the specification. Portions of these descriptions are reproduced or summarized below. Note that it is not Applicants' intention to limit the scope of the invention to what is described in this summary. This material is purely illustrative.

Figure 3, which is reproduced below for convenience, illustrates an electronic method for generating a configuration schema in accordance with the principles of the

present invention. The illustrated method can be used, for example, to generate an XML schema from the CLI commands associated with a Cisco™ router. With specific reference to independent claim 6, it recites accessing "a network component." In one embodiment for example, a system administrator (e.g. system administrator 125 depicted in FIG. 1) (in conjunction with an automated system) may connect to a network component (e.g., a router 120 depicted in FIG. 1) through, for example, a telnet connection. Next, the system administrator logs into the network component and activates a command extraction mode (steps 160 and 165) With regard to a Cisco™ router, the command extraction mode is activating by entering a "?" at the prompt. (See Specification Para. 0021; Page 9, lines 3-10).

Independent claim 6 also recites retrieving "a command set from the network component the command set including commands that the network component is capable of responding to." As described with reference to FIG. 3, a system administrator 125 may retrieve the primary commands, subcommands and bounds (steps 170, 175 and 180). This retrieval can be done through an automated, recursive search. For a Cisco™ router, the following search could be executed and the following results returned where ">" is the CLI prompt:

```
> ?
router
admin
global
> router?
bgp
ospf
>
```

This process could be repeated until termination for each command and subcommand. The output of a retrieval process, called a text file, for the "service" command is shown in Appendix A of Applicants' Specification (See Specification Para. 0021; Page 9, line 12-Page 10, line 3).

Once the commands, subcommands, and bounds are collected, they can then be recorded and cleansed (steps 185 and 190). Duplicate commands, for example, could be identified. When these duplicate commands include different subcommands and/or bounds, a single, cleansed command can be constructed to replace the duplicate commands.

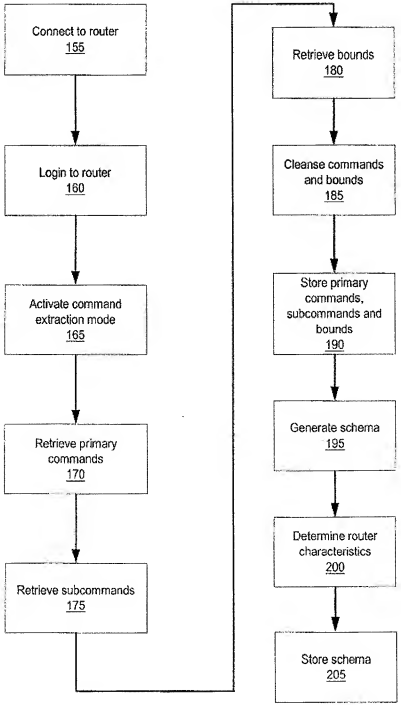
In addition, independent claim 6 recites generating "a configuration schema using the retrieved command set, wherein the generated configuration schema corresponds to the network component." As detailed in Applicants' Specification, for example, the cleansed commands, assuming that cleansing was necessary, can then be used to build a configuration schema, which in essence is a modeling of the router's command structure (step 195). An example snippet of such a modeling in an XML schema is represented by:

```
<xsd:element name="vlan">
    <xsd:complexType>
        <xsd:choice>
            <xsd:sequence>
                <xsd:element name="mapping">
            <xsd:complexType/>
        </xsd:element>
        <xsd:element name="dot1q"  fixed="dot1q">
            <xsd:complexType/>
        <xsd:element>
        <xsd:element name="ARG.001".
            <xsd:simpleType>
                .
                .
                .
```

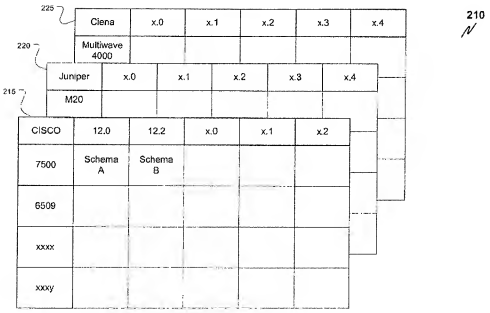
```
                </xsd:choice>
            </xsd:complexType>
        </xsd:element>
```

A more detailed example of an XML configuration schema is shown in Appendix B of Applicants' specification. The configuration schema in Appendix B corresponds to the "service" command, which is represented in Appendix A.

In one embodiment, the conversion between the text file, such as the one shown in Appendix A of Applicants' specification, and the XML configuration schema is performed by a Visual Basic program. This program identifies arrays of related commands in the text file. Individual arrays can be identified, for example, because they are generally separated by termination characters or by logical termination indicators. Additionally, when an input indicator is encountered in the text file, the program can insert a placeholder into the configuration schema to represent the input indicator. This placeholder can then be associated with the bounds for that particular input. For example, if the input corresponding to the input indicator should be between 1 and 10, a bound of 1 to 10 can be associated with the placeholder.


Connect to router
155
Login to router
160
Activate command extraction mode
165
Retrieve primary commands
170
Retrieve subcommands
175
Retrieve bounds
180
Cleanse commands and bounds
185
Store primary commands, subcommands and bounds
190
Generate schema
195
Determine router characteristics
200
Store schema
205

Independent claim 6 also recites determining "a characteristic of the network component, wherein the determined characteristic is indicative of at least one of: device type, manufacturer, model, and operating system version"; and storing "the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components." As detailed in Applicants' Specification, after the configuration schema has been generated, it is associated with characteristics of the router and stored accordingly (steps 200 and 205). For example, the configuration schema might be associated with a Cisco™ router, model 7500, OS version 12.0. A representation of a storage model 210 for storing configuration schema according to manufacturer, device type, device model, and OS version is shown in Figure 4, which is reproduced below. The first data block 215 is dedicated to Cisco™ routers as indicated in the upper left-hand corner. Each row represents a different model of Cisco™ device, and each column represents a different OS version. Similarly, the second data block is for Juniper™ routers 220 and the third is for Ciena™ devices 225 (Specification, Para. 24; page 11, line 14 to page 12, line 2).



Independent Claim 24

Although several embodiments of the present invention are disclosed in the specification, Figures 3 and 4 and the supporting text provides a good summary of embodiments which are exemplary of the subject matter defined by independent claim 24 and at least a portion of the dependent claims. The main text describing Figures 3 and 4 is located at paragraphs [0021]-[0024] (Page 9, line 3 to Page 12, line 2) of the specification. Portions of these descriptions are reproduced or summarized below. Note that it is not Applicants' intention to limit the scope of the invention to what is described in this summary. This material is purely illustrative.

Figure 3, which is reproduced below for convenience, illustrates an electronic method for generating a configuration schema in accordance with the principles of the present invention. The illustrated method can be used, for example, to generate an XML schema from the CLI commands associated with a Cisco™ router. With specific

reference to independent claim 24, it recites accessing "a network component." In one embodiment for example, a system administrator (e.g. system administrator 125 depicted in FIG. 1) (in conjunction with an automated system) may connect to a network component (e.g., a router 120 depicted in FIG. 1) through, for example, a telnet connection. Next, the system administrator logs into the network component and activates a command extraction mode (steps 160 and 165) With regard to a Cisco™ router, the command extraction mode is activating by entering a "?" at the prompt. (See Specification Para. 0021; Page 9, lines 3-10).

Independent claim 24 also recites retrieving "a command set from the network component the command set including commands that the network component is capable of responding to." As described with reference to FIG. 3, a system administrator 125 may retrieve the primary commands, subcommands and bounds (steps 170, 175 and 180). This retrieval can be done through an automated, recursive search. For a Cisco™ router, the following search could be executed and the following results returned where ">" is the CLI prompt:

> ?
> router
> admin
> global
> router?
> bgp
> ospf
>

This process could be repeated until termination for each command and subcommand. The output of a retrieval process, called a text file, for the "service" command is shown in

Appendix A of Applicants' Specification (See Specification Para. 0021; Page 9, line 12-Page 10, line 3).

Once the commands, subcommands, and bounds are collected, they can then be recorded and cleansed (steps 185 and 190). Duplicate commands, for example, could be identified. When these duplicate commands include different subcommands and/or bounds, a single, cleansed command can be constructed to replace the duplicate commands.

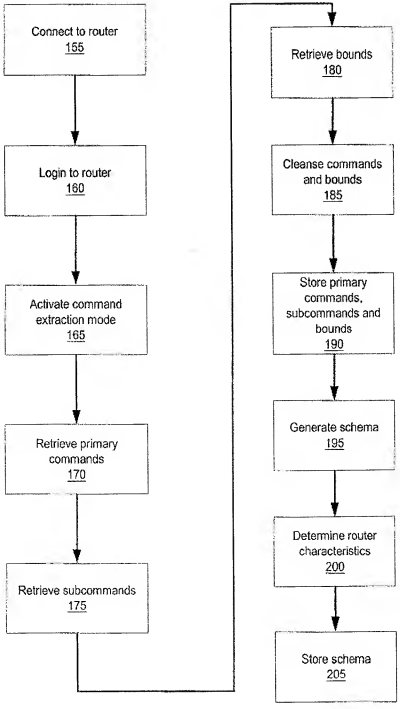
In addition, independent claim 24 recites generating "a configuration schema using the retrieved command set, wherein the generated configuration schema corresponds to the network component." As detailed in Applicants' Specification, for example, the cleansed commands, assuming that cleansing was necessary, can then be used to build a configuration schema, which in essence is a modeling of the router's command structure (step 195). An example snippet of such a modeling in an XML schema is represented by:

```
<xsd:element name="vlan">
        <xsd:complexType>
                <xsd:choice>
                        <xsd:sequence>
                                <xsd:element name="mapping">
                        <xsd:complexType/>
                </xsd:element>
                <xsd:element name="dot1q"  fixed="dot1q">
                        <xsd:complexType/>
                <xsd:element>
                <xsd:element name="ARG.001".
                        <xsd:simpleType>
                                .
                                .
                                .
                </xsd:choice>
        </xsd:complexType>
```

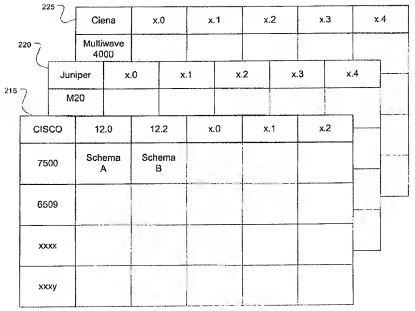
</xsd:element>

A more detailed example of an XML configuration schema is shown in Appendix B of Applicants' specification. The configuration schema in Appendix B corresponds to the "service" command, which is represented in Appendix A.

In one embodiment, the conversion between the text file, such as the one shown in Appendix A of Applicants' specification, and the XML configuration schema is performed by a Visual Basic program. This program identifies arrays of related commands in the text file. Individual arrays can be identified, for example, because they are generally separated by termination characters or by logical termination indicators. Additionally, when an input indicator is encountered in the text file, the program can insert a placeholder into the configuration schema to represent the input indicator. This placeholder can then be associated with the bounds for that particular input. For example, if the input corresponding to the input indicator should be between 1 and 10, a bound of 1 to 10 can be associated with the placeholder.


Connect to router
155
Login to router
160
Activate command extraction mode
165
Retrieve primary commands
170
Retrieve subcommands
175
Retrieve bounds
180
Cleanse commands and bounds
185
Store primary commands, subcommands and bounds
190
Generate schema
195
Determine router characteristics
200
Store schema
205

Independent claim 24 also recites determining "a characteristic of the network component, wherein the determined characteristic is indicative of at least one of: device type, manufacturer, model, and operating system version"; and storing "the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components." As detailed in Applicants' Specification, after the configuration schema has been generated, it is associated with characteristics of the router and stored accordingly (steps 200 and 205). For example, the configuration schema might be associated with a Cisco™ router, model 7500, OS version 12.0. A representation of a storage model 210 for storing configuration schema according to manufacturer, device type, device model, and OS version is shown in Figure 4, which is reproduced below. The first data block 215 is dedicated to Cisco™ routers as indicated in the upper left-hand corner. Each row represents a different model of Cisco™ device, and each column represents a different OS version. Similarly, the second data block is for Juniper™ routers 220 and the third is for Ciena™ devices 225 (Specification, Para. 24; page 11, line 14 to page 12, line 2).


225
Ciena
x.0
x.1
x.2
x.3
x.4
Multiwave 4000
220
Juniper
x.0
x.1
x.2
x.3
x.4
M20
215
CISCO
12.0
12.2
x.0
x.1
x.2
7500
Schema A
Schema B
6509
xxxx
xxxy
210

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 6-9, 11, 12, 16, 24, 25, and 27 are rendered unpatentable under 35 U.S.C. 103(a) over U.S. Patent No. 6,959,332 (Zavalkovsky) in view of U.S. Patent No. 6,816,897 (McGuire), and whether claim 10 is rendered unpatentable under 35 U.S.C. 103(a) over U.S. Patent No. 6,959,332 (Zavalkovsky) in view of U.S. Patent No. 6,816,897 (McGuire) and U.S. Patent Application 2003/0048287 (Little).

ARGUMENT

Applicants individually challenge the rejection of claims 6, 8-12, 24, and 27. These claims were not properly rejected. All other claims are allowable, at least, because they depend from allowable claims. A summary of the Examiner's position relative to claims 6 and 24 as well as the page number in this Appeal Brief where Applicant's remarks are found, is provided in the following table:

Limitations of claims 6 and 24	**Construct/Language in Zavalkovsky and McGuire identified by Examiner**	**Applicant's Response within this Appeal Brief**
retrieving a command set from the network component the command set including commands that the network component is capable of responding to	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Pages 10-12
generating a configuration schema using the retrieved command set	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Pages 12-13
storing the generated configuration schema in accordance with the determined characteristic	Nothing specifically identified but Col. 8, lines 1-21 of Zavalkovsky is cited	Page 13
so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components	Nothing specifically identified but McGuire at Col. 6, lines 16-35 is cited	Pages 13-14

A summary of the Examiner's position relative to dependent claims 8-12 and 27 as well as the page number in this Appeal Brief where Applicant's remarks are found, is provided in the following tables:

Limitations of claim 8	**Construct/Language in Zavalkovsky identified by Examiner**	**Applicant's Response within this Appeal Brief**
retrieving a set of primary commands	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Pages 14-15
retrieving a set of subcommands for each of the primary commands in the set of primary commands	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Pages 14-15
retrieving a set of bounds for a plurality of the set of subcommands for a first primary command	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Pages 14-15

Limitations of claim 9	**Construct/Language in Zavalkovsky identified by Examiner**	**Applicant's Response within this Appeal Brief**
identifying a command array in the command set, wherein the command array includes a primary command and a subcommand associated with the primary command	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Pages 15-16
extracting the primary command from the command array; and	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Pages 15-16
extracting the subcommand from the command array	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Pages 15-16

Limitations of claim 10	**Construct/Language in Zavalkovsky and Little identified by Examiner**	**Applicant's Response within this Appeal Brief**
forming an XML object using the extracted primary command and the extracted subcommand	The Final Action alleges Zavalkovsky teaches forming a generic object at Col. 7, line 56 through Col. 8, line 21 and Little teaches a CLI abstraction engine in which XML-based commands are translated to CLI-based commands for an embedded system, (page 1, paragraph 8), by means of a DTD-schema (page 4, paragraphs 63-65)	Page 16-17

Limitations of claim 11	**Construct/Language in Zavalkovsky identified by Examiner**	**Applicant's Response within this Appeal Brief**
the retrieved command set is a first command set and includes a plurality of primary commands	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Page 17-18
configuring the network component according to a first of the plurality of primary commands	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Page 18
retrieving a second command set	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Page 18
wherein the second command set includes a plurality of subcommands associated with the first of the plurality of primary commands and wherein the first command set and the second command set are different	Nothing specifically identified but Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky is cited	Page 18

Claim Limitations of claim 12	**Construct/Language in Zavalkovsky identified by Examiner**	**Applicant's Response within this Appeal Brief**
cleansing the retrieved command set	Nothing specifically identified but Col. 8, lines 13-21 of Zavalkovsky is cited	Page 19

Claim Limitations of claim 27	Construct/Language in Zavalkovsky identified by Examiner	Applicant's Response within this Appeal Brief
retrieve a bound for a first command in the command set	Nothing specifically identified but Col. 7, lines 56-67 of Zavalkovsky is cited	Page 19-20

Independent claims 6 and 24

Applicants submit that the 35 U.S.C. § 103(a) rejection against claims 6 and 24 is improper because there are several limitations in these claims that are neither taught nor suggested by Zavalkovsky and McGuire, and in addition, the Final Action has not identified with any specificity at least a suggestion of each claim limitation. Accordingly, the rejection against claims 6 and 24 should be withdrawn. For simplicity, claim 6 is directly addressed, but unless indicated otherwise, the same arguments apply to claim 24.

Claim 6 is reproduced below for convenience:

> An electronic method comprising:
> accessing a network component;
> retrieving a command set from the network component the command set including commands that the network component is capable of responding to;
> determining a characteristic of the network component, wherein the determined characteristic is indicative of at least one of: device type, manufacturer, model, and operating system version;
> generating a configuration schema using the retrieved command set, wherein the generated configuration schema corresponds to the network component; and
> storing the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components.

Zavalkovsky and McGuire neither disclose nor suggest "retrieving a command set from the network component"

The Final Action contends that Zavalkovsky teaches retrieving a command set from the network component at Col. 7, lines 56-67. Applicants disagree. As recited in claim 6, the claimed "command set" is used to generate "a configuration schema." But Zavalkovsky does not teach retrieving anything that can be used to generate a "configuration schema." At most, Zavalkovsky teaches determining CLI commands based upon *a current configuration* of their device. Specifically, at Col. 7, lines 56-67 Zavalkovsky discloses:

> [t]he current configuration of each device is received and analyzed. Current device configuration information may be obtained using a special CLI command (e.g., "show running config" on a Cisco router), or by other conventional means, such as device discovery processes that use one or more SNMP query messages to obtain MIB variable values....based on the current device configuration information received from the device, *the process determines one or more specific CLI commands that would create such configuration* if sent to and executed by the operating system of the device. As a result, a list of CLI commands for the current device configuration is created and stored (emphasis added).

As Applicants' specification teaches, the claimed configuration schema is, in essence, a modeling of a network component's command structure (See Applicants' Specification, Para. [0022]). Applicants submit that Zavalkovsky's list of CLI commands for a current device configuration is neither intended to generate a configuration schema nor is the list sufficient to generate a configuration schema. In particular, Zavalkovsky's list of CLI commands is merely a list of commands for a current configuration of a device, but such a list does not provide adequate information to model the command structure of a network device. As a consequence, Zavalkovsky's list of CLI commands at Col. 7, lines

56-67 can not correspond to the recited retrieved "command set;" thus the rejection of claims 6 and 24 is improper.

Zavalkovsky and McGuire neither disclose nor suggest "generating a configuration schema using the command set"

The Final Action contends that Zavalkovsky also teaches, at Col. 7, line 56 through Col. 8, line 21, "generating a configuration schema using the retrieved command set." Again Applicants disagree. Zavalkovsky simply does not teach generating a configuration schema. A simple word search further supports this conclusion because neither "configuration schema," nor even "schema" appear in Zavalkovsky.

Applicants also submit that the rejection of claims 6 and 24 is also improper for failing to identify with any specificity the constructs within Zavalkovsky that allegedly correspond to the claimed "configuration schema." In particular, although the Final Action does mimic back the claim language, it does not identify any specific language or constructs within Zavalkovsky that allegedly teaches a "configuration schema."

More problematically, as discussed further herein, the Final Action alleges that Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky teaches a "configuration schema," a "command set," "primary commands," "subcommands," a "command array," "a second command set," and a "set of bounds;" yet the Final Action does not specifically identify any language nor any constructs in Col. 7, line 56 through Col. 8, line 21 of Zavalkovsky that teaches these claimed limitations. Moreover, Zavalkovsky teaches many different constructs in Col. 7, line 56 through Col. 8, line 21; thus the Examiner has failed to honor Rule 37 CFR 1.104 (c)(2), which requires that, for references like Zavalkovsky, "the particular part relied on must be designated as nearly as practicable."

Zavalkovsky neither discloses nor suggests "storing the generated configuration schema"

The Final Action also alleges that Zavalkovsky teaches, at Col. 8, lines 1-21, "storing the generated configuration schema." Again, Applicants disagree. Zavalkovsky does not teach a configuration schema at all, and as a consequence, Zavalkovsky can not teach storing a configuration schema.

The Final Action does not identify with any specificity the construct in McGuire that allegedly corresponds to the "configuration schema."

The Final Action alleges that McGuire teaches, at Col. 6, lines 16-35, storing the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components. The Final Action, however, does not specifically identify any construct among the many items disclosed in this portion of McGuire that allegedly corresponds to the claimed "configuration schema;" thus the rejection of claims 6 and 24 is also improper for this additional reason, and accordingly, the rejection against claims 6 and 24 should be withdrawn.

Applicants also submit dependent claims 7-12, 16, 25 and 27 are also allowable, at least, by virtue of being dependent from allowable independent claim 6 or 24.

Dependent claim 8

Dependent claim 8 recites:

> 8. The method of claim 6, wherein retrieving the command set comprises:
> retrieving a set of primary commands;
> retrieving a set of subcommands for each of the primary commands in the set of primary commands; and
> retrieving a set of bounds for a plurality of the set of subcommands for a first primary command.

The Final Action contends that Zavalkovsky teaches retrieving a set of primary commands at Col. 7, lines 56-67, but the Final Action does not provide any specificity as to what commands disclosed by Zavalkovsky allegedly correspond to the primary commands.

In addition, the Final Action contends that Zavalkovsky also teaches retrieving a set of subcommands for each of the primary commands at Col. 7, lines 56-67. Applicants again disagree. Zavalkovsky does not suggest anything about a set of subcommands for each of the primary commands. For example, a simple word search of Zavalkovsky reveals that the "subcommands" limitation is not found at all in Zavalkovsky; thus the rejection is improper and should be withdrawn.

The Final Action also contends that the same twelve lines of Zavalkovsky (i.e., Col. 7, lines 56-67) also teaches "retrieving a set of bounds for a plurality of the set of subcommands for a first primary command." Applicants disagree. Zavalkovsky does not teach retrieving a set of subcommands from a network component and Zavalkovsky certainly does not teach retrieving a set of bounds for a plurality of the set of subcommands. A simple word search of Zavalkovsky, for example, makes clear that the

"bounds" limitation is not found within Zavalkovsky. Accordingly the rejection is improper and should be withdrawn.

Moreover, the Final Action fails to make out a proper rejection because the Final Action does not provide any specificity as to what constructs within Zavalkovsky allegedly correspond to the claimed "subcommands" nor any specificity as to what constructs in Zavalkovsky correspond to the "set of bounds" recited in claim 8. As a consequence, the rejection itself is improper and should be withdrawn.

Dependent claim 9

Dependent claim 9 recites:

> The method of claim 8, wherein generating the configuration schema comprises:
> identifying a command array in the command set, wherein the command array includes a primary command and a subcommand associated with the primary command;
> extracting the primary command from the command array; and
> extracting the subcommand from the command array.

The Final Action contends that Zavalkovsky teaches, at Col. 7, lines 56 through Col. 8, line 21, "identifying a command array in the command set." Applicants disagree. Applicants have reviewed not only Col. 7, lines 56 through Col. 8, line 21, but Zavalkovsky as a whole, and there is no suggestion of identifying a command array in a command set. A simple word search of Zavalkovsky, for example, reveals that the "array" limitation does not appear at all in Zavalkovsky.

The Final Action also alleges Col. 7, lines 56 through Col. 8, line 21 of Zavalkovsky teaches extracting the primary command from the command array and

extracting the subcommand from the command array. Again Zavalkovsky does not teach a command array at all; thus Zavalkovsky can not possibly teach extracting the primary command from the command array and extracting the subcommand from the command array.

Moreover the Final Action fails to provide the requisite specificity when rejecting claim 9. In particular, the Final Action does not identify a single construct within Zavalkovsky that allegedly corresponds to the recited "command array." Thus, not only does the rejection fail because the prior art does not at least suggest each limitation, the rejection itself is improper for lacking the requisite specificity. Accordingly, the rejection against claim 9 should be withdrawn.

Dependent claim 10

Claim 10 stands rejected under 35 U.S.C. 103(a) over U.S. Patent No. 6,959,332 (Zavalkovsky) in view of U.S. Patent No. 6,816,897 (McGuire) and U.S. Patent Application 2003/0048287 (Little). Claim 10 recites:

> 10. The method of claim 9, wherein generating the configuration schema comprises:
> forming an XML object using the extracted primary command and the extracted subcommand.

The Final Action contends that Zavalkovsky teaches, at Col. 7, line 56 through Col. 8, line 21, the formation of a generic object using the extracted primary command and extracted subcommand. Again, Zavalkovsky does not teach extraction of any commands, nor does Zavalkovsky disclose any "primary commands" or "subcommands."

In addition, Zavalkovsky does not teach nor suggest forming a generic object as the Final Action alleges. A simple word search for example reveals that neither "generic

object" nor "generic" is found in Zavalkovsky. Moreover, the rejection is improper because it does not identify—with any specificity—what construct within Zavalkovsky is the alleged generic object.

In addition, the Final Action alleges that a modification of the teachings of Zavalkovsky with Little renders an XML-based generic object. Applicants disagree. Again, Zavalkovsky does not teach any generic objects that can be modified, and in addition, the Final Action does not state a clear basis for the conclusion that Little's command line interface abstraction engine renders obvious the claimed XML object that is formed using the extracted primary command and the extracted subcommand. As a consequence, the rejection is improper and should be withdrawn.

Dependent claim 11

Claim 11 recites:

> The method of claim 6, wherein the retrieved command set is a first command set and includes a plurality of primary commands and wherein generating the configuration schema comprises:
> configuring the network component according to a first of the plurality of primary commands; and
> retrieving a second command set;
> wherein the second command set includes a plurality of subcommands associated with the first of the plurality of primary commands and wherein the first command set and the second command set are different.

The Final Action contends that Zavalkovsky teaches, at Col. 7, lines 56 through Col. 8, line 21, the retrieved command set is a first command set and includes a plurality of primary commands and configuring the network component according to a first of the plurality of primary commands. Zavalkovsky does not teach configuring the network component according to a first of the plurality of primary commands. A word search of

Zavalkovsky, for example, reveals that the "primary" limitation is not found at all in Zavalkovsky.

In addition, the Final Action alleges that Zavalkovsky teaches retrieving a second command set at Col. 7, lines 56-67. As discussed in the arguments above relative to claims 6 and 24, which are incorporated herein be reference, Zavalkovsky does not teach retrieving any command sets. As a consequence, Zavalkovsky can not teach retrieving a second command set.

Moreover, the claimed second command set "includes a plurality of subcommands associated with the first of the plurality of primary commands wherein the second command set includes a plurality of subcommands associated with the first of the plurality of primary commands and wherein the first command set and the second command set are different." Zavalkovsky does not suggest anything remotely similar to these limitations. Again, the "primary" and "subcommands" limitations are nowhere to be found when word-searching Zavalkovsky.

Moreover the Final Action fails to specifically identify any constructs that allegedly correspond to the "first command set" and the "second command set" wherein the first command set and the second command set are different. As a consequence, the rejection is improper because the prior art fails to suggest each limitation of claim 11 and because the rejection is itself is deficient for wholly lacking specificity. In particular, the Final Action mimics back the claim language and alleges all the limitations are found within Col. 7, line 56 through Col. 8, line 21, but does not cite even one word from this portion of Zavalkovsky that allegedly teaches the claimed limitations. Accordingly, the rejection against claim 11 should be withdrawn.

Dependent claim 12

> The method of claim 6, further comprising:
> cleansing the retrieved command set.

The Final Action alleges that Col. 8, lines 13-21 Zavalkovsky teaches cleansing the retrieved command set. Again, Zavalkovsky does not teach retrieving the claimed "command set" from a network component, so Zavalkovsky can not possibly teach cleansing the claimed command set. Moreover, Zavalkovsky does not teach cleansing--a simple word search of Zavalkovsky reveals that the "cleansing" limitation does not appear in Zavalkovsky at all.

Moreover, the Final Action fails to identify what construct within Zavalkovsky allegedly corresponds to the claimed "retrieved command set" and fails to identify what specific teaching within Zavalkovsky allegedly teaches "cleansing." As a consequence, the rejection is improper because the prior art fails to suggest each limitation of claim 12 and because the rejection is itself deficient for wholly lacking any specificity. Accordingly, the rejection against claim 12 should be withdrawn.

Dependent claim 27

Claim 27 recites:

> The computer program product of claim 24, wherein the plurality of instructions are further configured to instruct the electronic device to:
> retrieve a bound for a first command in the command set.

The Final Action Alleges that Zavalkovsky teaches, at Col. 7, lines 56-67 "retrieve a bound for a first command in the command set." Applicants disagree. Zavalkovsky does not teach retrieving a bound for a first command. A simple word

search of Zavalkovsky, for example, makes clear that the "bound" limitation is not found within Zavalkovsky.

Moreover, the Final Action fails to make out a proper rejection because the Final Action does not provide any specificity as to what language within Zavalkovsky allegedly teaches the "bound" limitation. As a consequence, the rejection itself is improper for failing to provide the requisite specificity, and the rejection does not cite prior art that at least suggests each limitation of claim 27. Accordingly, the rejection against claim 27 should be withdrawn.

SUMMARY

All of the pending claims are patentable for the reasons set forth herein, and Appellant respectfully requests such finding.

COOLEY GODWARD KRONISH LLP
ATTN: Patent Group
777 6th Street, NW
Suite 1100
Washington, DC 20001
Tel: (720) 566-4044
Fax: (202) 842-7899

Respectfully submitted,
COOLEY GODWARD KRONISH LLP

By: Thomas M. Croft
Thomas M. Croft
Reg. No. 44,051

CLAIMS APPENDIX

Claims 1-5 Cancelled

6. An electronic method comprising:

accessing a network component;

retrieving a command set from the network component the command set including commands that the network component is capable of responding to;

determining a characteristic of the network component, wherein the determined characteristic is indicative of at least one of: device type, manufacturer, model, and operating system version;

generating a configuration schema using the retrieved command set, wherein the generated configuration schema corresponds to the network component; and

storing the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components.

7. The method of claim 6 further comprising:

activating a command extraction mode of the network component.

8. The method of claim 6, wherein retrieving the command set comprises:

retrieving a set of primary commands;

retrieving a set of subcommands for each of the primary commands in the set of primary commands; and

retrieving a set of bounds for a plurality of the set of subcommands for a first primary command.

9. The method of claim 8, wherein generating the configuration schema comprises:

identifying a command array in the command set, wherein the command array includes a primary command and a subcommand associated with the primary command;

extracting the primary command from the command array; and

extracting the subcommand from the command array.

10. The method of claim 9, wherein generating the configuration schema comprises:

forming an XML object using the extracted primary command and the extracted subcommand.

11. The method of claim 6, wherein the retrieved command set is a first command set and includes a plurality of primary commands and wherein generating the configuration schema comprises:

configuring the network component according to a first of the plurality of primary commands; and

retrieving a second command set;

wherein the second command set includes a plurality of subcommands associated with the first of the plurality of primary commands and wherein the first command set and the second command set are different.

12. The method of claim 6, further comprising:

cleansing the retrieved command set.

Claims 13-15 (cancelled)

16. The method of claim 6, wherein accessing a network component comprises:

accessing a router.

Claims 17-23 (cancelled).

24. A computer program product comprising:

a computer readable storage medium; and

a plurality of instructions stored upon the storage medium, the plurality of instructions configured to instruct an electronic device to:

access a network component;

retrieve a command set from the network component the command set including commands that the network component is capable of responding to;

determine a characteristic of the network component, wherein the determined characteristic is indicative of at least one of: device type, manufacturer, model, and operating system version; ~~and~~

generate a configuration schema corresponding to the network component, wherein the configuration schema is based upon the retrieved command set; and

store the generated configuration schema in accordance with the determined characteristic so as to enable the configuration schema to be identified from among a collection of configuration schemas that includes configuration schemas that are associated with other network components.

25. The computer program product of claim 24, wherein the plurality of instructions are further configured to instruct the electronic device to:

activate a command extraction mode associated with the network component.

26. (cancelled).

27. The computer program product of claim 24, wherein the plurality of instructions are further configured to instruct the electronic device to:

retrieve a bound for a fist command in the command set.

Claims 28-29 (Cancelled)

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None